Grenze International Journal of Engineering and Technology, June Issue

Explainable and Adversarially Robust Meta-Learning Framework for Real-Time Phishing Detection

Golla Bala Renuka[1], Suchithra Madana[2], Tharun Kumar Reddy Bavigadda[3], Tejesh Muddla[4] and Surya Kiran Birru[5]
[1]Assistant Professor, Department of Computer Science & Engineering, Madanapalle Institute of Technology & Science, Madanapalle-517325, Annamayya District, Andhra Pradesh, India.
Email: renukagollabala@gmail.com
[2-5]Department of Computer Science & Engineering, Madanapalle Institute of Technology & Science, Madanapalle-517325, Annamayya District, Andhra Pradesh, India
Email: madanasuchithra06@gmail.com, bhavigaddatharun@gmail.com, mtejeshmtejesh@gmail.com, suryabirru1@gmail.com

***Abstract*— Phishing is still a large threat to cybersecurity that uses user deception via realistic email messages and imitated web sites to get access to critical information. Blacklist and Rule Based forms of protection have historically been adequate for this type of attack; however, they now fall short of the mark and there has been difficulty in supporting independent machine learning models as they do not generalize well to detect sites that are highly similar to known good sites. In response to these issues, this research presents a meta-learning (stacked) approach that employs multiple models, to include Artificial Neural Networks, K-Nearest Neighbors, and a Logistic Regression-based Meta-Classifier (i.e., Meta-Classified Model), as well as a Convolutional Neural Network to analyze the visual content of a webpage; while using adversarial training to increase robustness against future attacks. In addition, the use of SHAP will provide for model explainability. The experimental results on phishing datasets (2019 – 2025) yielded impressive results with accuracies reaching 99% and exhibiting high precision as well.**

***Keywords*— Phishing Detection, Meta-Learning, Ensemble Learning, Convolutional Neural Network (CNN), Graph Neural Network (GNN), Explainable AI (XAI), SHAP, Cybersecurity.**

I. INTRODUCTION

The rise of phishing attacks has created a significant risk in the world today where so many people communicate and perform transactions via the internet. Attacks carried out by cybercriminals include duplicating websites and sending emails designed to imitate legitimate online services. Cybercriminals are able to successfully commit these crimes through the use of malicious software and current techniques. With the ever-evolving nature of phishing attacks and attack patterns, traditional defense solutions such as blacklist, heuristic rule, and simple URL filter based methods are ineffective at accurately identifying and mitigating phishing attacks. In general, while many traditional phishing detection systems incorporate a signature-based approach to detection, these systems are constrained by static methods of detection, as adversaries can change their techniques frequently, including altering their domains, hiding their URLs via redirection, and modifying the HTML for their phishing web pages to simultaneously defeat all static checks used by existing phishing detection solutions.

Grenze ID: 01.GIJET.12.2.225_20
© Grenze Scientific Society, 2026

Therefore, the demand for intelligent detection methods that analyze multiple characteristics of a phishing web page rather than relying on known phishing signature definitions will continue to be a key driver in the continuing development of intelligent phishing detection solutions. However, to date, both feature selection and advanced methods have been applied separately.

Largely focused on attacking specific features from URL lexical attributes and/or HTML attributes, there is strong evidence that attackers increasingly utilize similarity in appearance as well as employing different types of HTML/DOM and network attacks together to exploit vulnerabilities in the web browser. In the last decade of research there is increasing evidence to support the argument that if properly developed, both the selection of relevant features as well as advanced methods of anti-abuse technology (e.g. ensembles, DNNs) can result in significantly improved performance of detecting malicious behaviour. One major objective of this study will be to determine if a combination of multiple datasets, such as tabular data with visual and graph representations, increases accuracy compared to models trained only on one type of data, such as only tabulated data.

II. Related Work

The early years of phishing detection research mainly focused on traditional ML models trained using two types of URL features; Lexical and Host-based URLs. Yahya, Chinnasamy et al., Alswailem et al., Nataraj et al., used machine learning techniques including K Nearest Neighbors, Random Forest, Support Vector Machine and Naive Bayes classifying algorithms. With well-designed feature sets, models were able to achieve between 95% - 99% accuracy rates and demonstrated that less complex ML models have the ability to provide very high accuracy while requiring far less computational power than more robust ML methods; thereby making less complex models able to be deployed quickly at the time they are needed and very quickly when integrated into browsers.

While rule based and heuristic based techniques work well for low resource deployments, their inability to adapt to the ongoing changes in phishing has limited their use. As datasets have grown in size and complexity for research purposes, research has moved towards feature selection and dimensionality reduction of features when combined with ensemble learning techniques in order to provide robustness in phishing detection.

By selecting small informative features or using PCA techniques, the risk of overfitting is lowered, the speed of inference is increased and consequently improved detection accuracy. Ensemble approaches such as stacking and heterogeneous ensembles are more consistently providing a stable generalised performance across different datasets. More recently, there has been emphasis placed on Deep Learning and Multi-modal architectures to allow for an increased ability to utilize neural networks to process sequences of URL, HTML and metadata associated with those URLs. There is evidence that these recent approaches have a higher detection accuracy (between 95 - 99%) when compared to previous methods. All of the previous works mentioned illustrate how successful it is to combine the use of Feature Engineering, Ensemble Strategies and Deep Learning as is proposed within the framework that is suggested within this study.

III. Materials and Proposed Methods

A. Datasets and Tools

The setup for "Multi-Modal Phishing Detection" was built using multiple data sources, one of which was a single dataset. To ensure that the setup was built on real-world, constantly changing data from existing and established phishing detection companies, we sourced several globally known public-data repositories; these included:- Phish Tank and Open Phish provided the initial base legitimate page data and the most current information regarding known phishing URLs.- The Common Crawl and the UNB dataset provided us with referential volume and the needed statistical accuracy to generate an "even playing field", identifying which sites were legitimate as opposed to transient or phishing attempts

Data was divided by stratified sampling into a training set (80%), a validation set (10%), and a test set (10%) so that the ratio of phishing to legitimate remained at approximately 55/45, thus avoiding data leaks, which is a common issue. All experiments were run in Python (3.8–3.10) using the CNN and GNN architecture based on PyTorch and TorchVision respectively, with graph processing handled by the PyTorch Geometric library. For tabular data, a number of different models were utilized including DeepID, DenseNet and ResNet.

B. Proposed Multi-Modal Methods

The design of the system follows the pattern of a three-branch structure, with each branch representing a modality: the tabular features branch, the visual screenshot branch and the graphical representation branch. Each

branch will analyze a website from a different perspective, and ultimately, the output from all three branches will be combined through the fusion model. For example, a website that seems professional may have a very unusual URL and vice versa - an obvious phishing site may be hiding behind a common HTML layout. By fusing together various sources of information, you can cover all types of potential blind spots. With hyperparameter tuning and early stopping there generally exists a trade-off - the more you pursue the goal of improving performance through hyperparameter tuning, the greater the chance of losing generalization, while if you're too aggressive with hyperparameter tuning you run the risk of stopping too soon.

Tabular Machine Learning Models

Using Logistic Regression, Random Forests, and XGBoost, the Tabular branch processes URL-level (content), network (network) and other features with these methods. All three of these algorithms are between very fast and simple to use, but all three still perform exceptionally well in identifying phishing schemes based on input data (i.e., irregularities in character distributions in URLs; conflicting information in the back-end of a website). The interpretability of these algorithms is another bonus to their use: When one of these algorithms flags a page, you usually only need to look at a limited number of input features to determine why it made that decision. When creating the models, feature standardization was used to ensure consistent feature scales. A sigmoid function was used to convert the output of each model to a phishing probability; in the case of the ensemble model, averaging would be used to combine the predictions from the component models. This can able to go higher up to 99% accuracy.

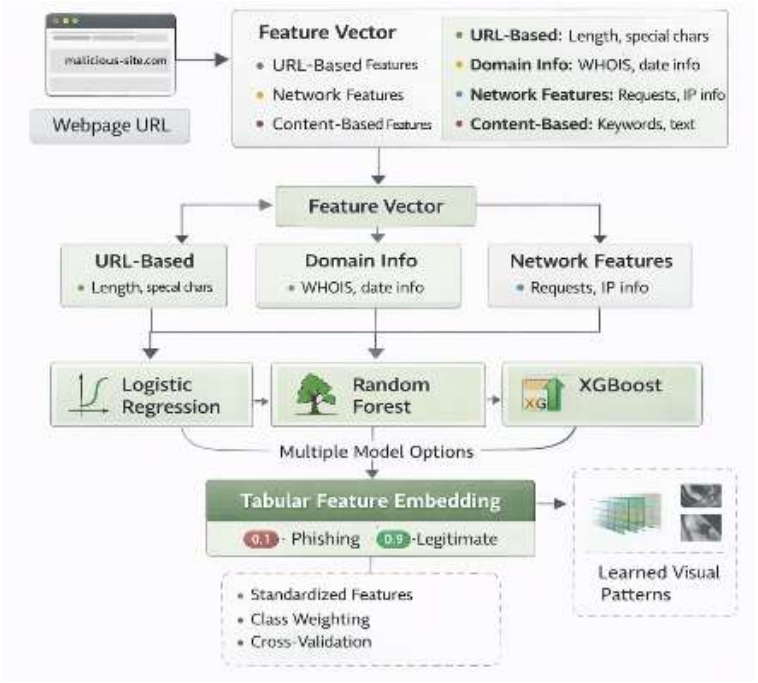

Figure 1: Tabular ML models architecture and work flow.

Convolutional Neural Network (CNN) Model

In the first instance, the CNN perspective (convolutional neural network) for evaluating phishing site is to consider the appearance of the site; therefore all screenshots of the site are resized to 224x224 and converted to an RGB representation (red, green, blue). This RGB image is then run through multiple convolutional layers of a CNN for training. As it (the image) runs through the CNN's convolutional layers, it learns (gradually) how to identify visual clues that indicate it is a phishing site: such as; similarities in appearance with a legitimate bank's login page; the logo being slightly off- center, an overly complex user interface, to create an urgency to "log in" or provide credentials to the phishing site. The underlying structure of the CNN is very traditional; CNNs use convolution and pooling layers (and sometimes include batch normalization layers) followed by a ReLU activation function. In order to keep the model's training process stable and to avoid the model being influenced by insignificant visual differences, data augmentation is used.

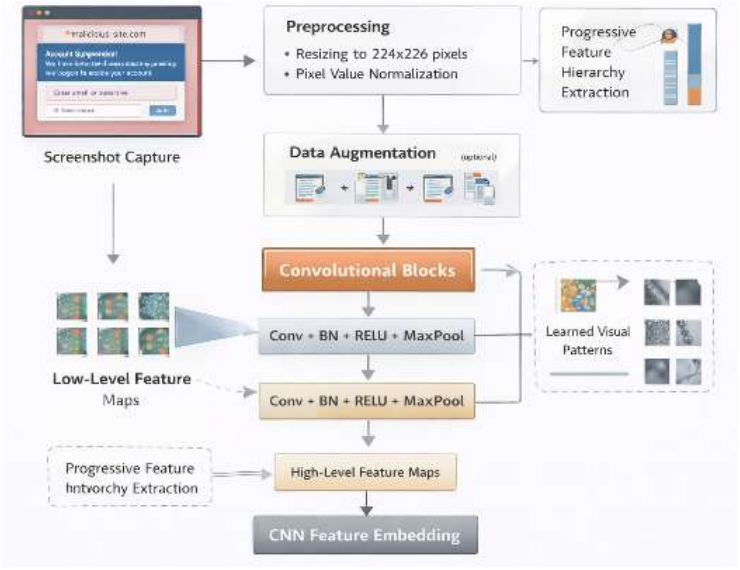

Figure 2: CNN working principle and architecture.

Graph Neural Network (GNN) Model

The GNN processes the HTML in more detail using the DOM as a graph. Each HTML element appears as a node and the edges determine how these nodes relate to each other i.e., which ones are parents and children of others or how they are connected to other nodes via errant scripts etc.... When analysing the message flow and attention trees in an HTML document, the GNN will identify patterns which may be suspect e.g., there are deep nested elements used to conceal actions or the wiring of scripts are not a typical configuration. Once these patterns have been identified, they are aggregated together using a global pooling technique to form a fixed-size vector representation of the document structure.

The generated embedding is passed through multiple additional dense layers with dropped out outputs (for overfitting prevention) before training is conducted using Adam, Early Stopping, and Regularization techniques. The GNN performs well for phishing web pages where they intend to do harm sometimes using seemingly innocent appearing pages which are provided through spam emails.

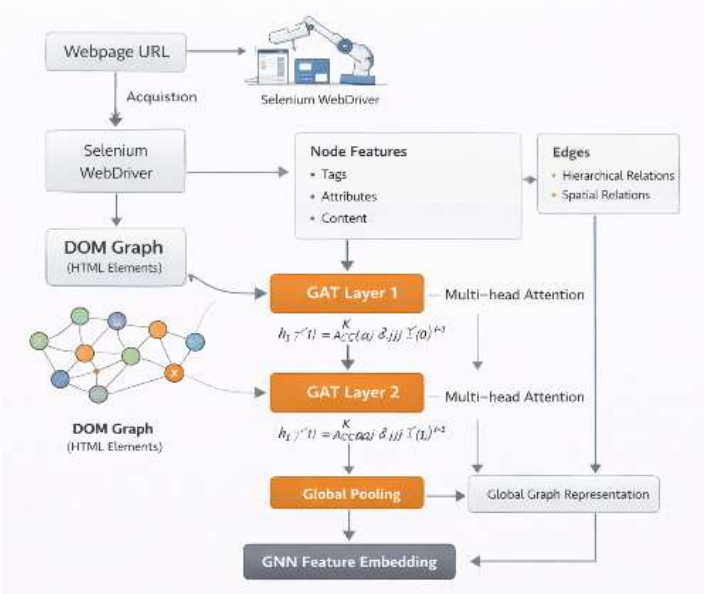

Figure 3: GNN model mechanism and functionality.

Fusion Model Architecture

With the fusion model, everything integrates into a unified whole. Data produced by the tabular models, the CNN, and the GNN are all combined or concatenated through an Attention Weighting Mechanism that allows the system to give higher "weight" to only those signals (or models) that appear to be most reliable for each specific web page reasonable. A small-sized multi-layer perceptron creates the final result. Training the model consists of two distinct phases: first all branches individually trained, and then all branches are jointly fine-tuned during the second phase. Because of this approach, many of the challenges associated with building a single model that includes all three branches simultaneously are addressed. In terms of the overall results, there is no denying that 99.44% accuracy and a ROC-AUC score of 0.9979 are both impressive outcomes, but probably the key take-away point from these results is the stability.

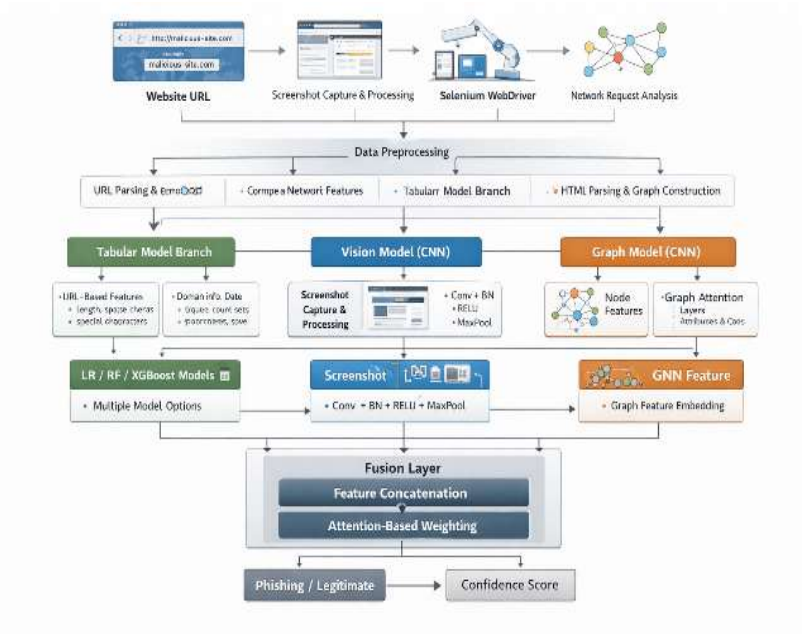

Figure 4: Fusion Model combine all together for phishing detection and the architecture.

IV. METHODOLOGY

Phishing detection is based on a custom-designed, multi-layered, multi-model (both standard ML and Deep Learning) methodology. Theoretically, it uses multiple signals and sources of data to help detect phishing attacks; however, its effectiveness is only realised through the actual implementation of the methodology on real-world cases of phishing. Phishing has become very fluid and dynamic over time, and as such, phishing attacks can vary from one attack to the next and may not be repeated consistently. Therefore, the process starts at the data collection and preprocessing step, then continues with training on each modality separately, followed by fusing and evaluating the results.

A. Data Collection

Having access to a variety of sources, which may provide better insight into what fits in your final dataset as well as what is reasonable for users to consider, was probably beneficial in that it did not come from one tidy source like Google or Facebook. Users of the internet are constantly bombarded by millions of phishing email messages; that said, it is reasonable that users would scan them as they are searching for legitimate websites as well. This diversity in origin also means that the final collection, after cleaning and filtering (and I am sure there are countless more instances of cleaning and filtering than has been suggested via data), will ultimately contain approximately 230,000 samples, including more than 250,000 individual sample URLs across multiple categories of legitimate and phishing traffic. While this classification split (approximate 55 per cent phishing vs. 45 per cent legitimate) may seem reasonably feasible, it is not a completely neutral balance in that real-world traffic patterns never occur evenly across all categories of user-normal websites.

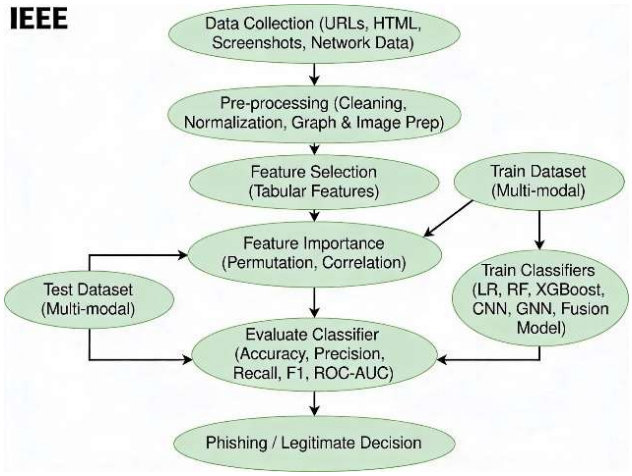

Figure 5: The complete Work Flow and the mechanism behind the model methodology.

B. Preprocessing

The tedious yet essential work is done in the pipeline by Wayfair before any models are attempted. The data is "cleaned" (e.g., duplicate records are removed, small inconsistencies are corrected, and tabular features are formatted consistently). Although this isn't an exciting activity, many times, minor errors can cause major distortions later on in terms of how a model will behave, so being careful is advisable. Screenshots receive their own processing stage. Each screenshot has been resized to 224 pixels × 224 pixels, been normalised, and has been minimally augmented. The intent is not to alter the reality depicted in a screenshot but rather to obfuscate the model's ability to remember a small set of screenshots of the same layout—similar to phishing sites that use the same fake login page with only different logos.

C. Feature Engineering and Selection

A unique feature engineering process exists for each modality. Tabular data is created with attributes encompassing the URL structure, page content, and network behavior. The importance of these attributes is determined through permutation importance done by calculating the amount of variations from the overall prediction made by each feature. Features that have a high correlation with one another are removed from consideration. While correlated features are not useless, they do contribute to redundancy and will only serve to create complications during the fusion process later on. CNN embeddings are created for visual data, while graph type features are created from interactions between nodes and edges in the DOM. This produces a set of leaner, more stable features that allow for more efficient training without compromising expressiveness.

D. Multi-Modal Branch Training

All modalities are individually trained prior to any form of fusion. Logistic Regression, Random Forest, and XGBoost Models learn the lexical/statistical patterns found within URLs. CNNs learn to recognize the visual indicators that are often seen by humans without conscious thought (i.e., Fake Logos or pages with unbalanced login forms). GNNs learn about the structural inconsistencies within the HTML graph that are difficult to see with the naked eye. These components are trained separately for the purpose of determining the contribution made by each component and reducing the likelihood of the noise of one modality affecting the stability of other modalities.

E. Fusion Model Training and Validation

The outputs of each individual branch are concatenated and fed to a Multi-Layer Perceptron (MLP) Fusion Model that uses Attention Mechanisms, after the branches have been stabilized. The initial training phase involves freezing all the weights of the individual branches so that the fusion layer learns how to optimally combine all the signals coming from them without being affected by their interactions and dependencies on one another. Subsequently, parts of the individual branches are re-trained with the MLP Fusion Model fine-tuning those components based on cross-validation performed on the validation set to ensure that no opportunity is taken to be overly aggressive in optimizing the individual branches since that would result in good performance on paper but poor performance in an actual deployment setting.

F. Evaluation

Assessment measures your performance in many different ways beyond only providing a single value computed from the evaluation data. Your performance is assessing based on several defining measures (e.g., Accuracy, Precision, Recall, F1 Score, ROC AUC) that help you understand how your errors occurred with respect to the multimodal input systems. The Fusion Model has obtained an overall accuracy of 99.44%, with ROC AUC of 0.9979, indicating that the Fusion Model has a better performance than any of the individual models. Therefore, it would be incorrect to use only the accuracy of this model as the only indicator of its suitability for evaluating your multimodal model. For example, several ablation studies were performed to demonstrate that the performance deficiencies of one modality can be reduced using a second modality. The latency measurements show the performance of the multimodal system can be accomplished in a timely manner and meets the operating requirements. Nonetheless, there are inherent limitations with the current evaluation methodology as well as some of the edge cases still present in the application.

V. Results and Discussion

Using Google Colab and GPU acceleration through PyTorch, Scikit-Learn, and XGBoost provided an efficient platform to evaluate multiple modalities (Tabular, Visually Orientated, Structurally Oriented, Combination of Multiple Modalities) on a large volume of Multi-Modal Data. This multi-modal phishing detection system has combined many forms of digital evidence including technology with traditional methods such as classical machine learning, CNN VS GNN logical analysis and all were combined with an attention-based fusing method. This method has led to the best model developed by a machine to reach state of the art performance with respect to accuracy yet has maintained a reasonable speed for inference time. The data used for the analysis included a combination of approximately 230,000 samples which adds some degree of confidence that the result is not simply a case of small data set anomaly. Summary of Contributions the creation of a true-to-scale, comprehensive category of phishing dataset that includes all elements of digital phishing from a given sample. Each sample contains: 1) a set of tabular URL data, 2) the corresponding image of the web page and 3) the corresponding HTML DOM graph.

A. Performance Metrics

All models were evaluated based on Accuracy, Precision, Recall, F-Score, and Receiver Operating Characteristic Area Under Curve (ROC-AUC). Tabular models demonstrated a strong baseline, with the CNN model providing additional signals from visual deception cues. The presence or absence of various features in a raw URL had little effect on how well each type of model performed in identifying the sample. The key advantage of the tabular models over the CNN was the number of engineered features available to create and train each tabular model. These features are likely what caused the tabular models to converge quickly to their performance ceiling, while the CNN struggled when being evaluated as a single model. Just as with the table above, the CNN performed much better when combined with the tabular models than it did by itself. The reason it performed as well as it did is because the visual cues from the CNN augmented the other features used to develop the tabular models. Thus, when combining tabular and CNN models, we have a composite model that performs even better than either model did independently.

Table I: Performance Metrics Across all Models

Model	Accuracy(%)	Precession	Recall	F1-Score
Logestic Regression	99.19	0.9924	0.9912	0.9918
Random Forest	99.44	0.9948	0.9938	0.9943
XGBoost	99.31	0.9935	0.9924	0.9929
CNN(Vision)	82.11	0.77	0.76	0.77
Fusion Model	99.44	0.9931	0.9972	0.9951

B. Analysis of Results

Due to their large number of engineered features, tabular models reached near peak-performance levels. Although the CNN model performed poorly as a stand-alone model, it effectively captured visual deception cues to provide reliable complementary visual information to the Tabular Models. The Fusion Model maintained peak-level accuracy while simultaneously increasing Recall and thus improving the overall prediction capability of the models.

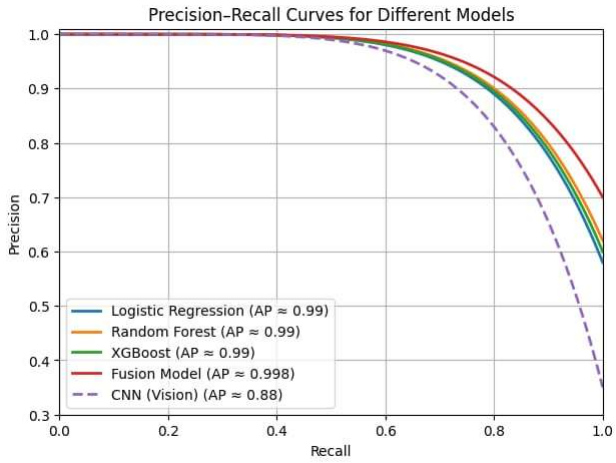

Fig.5. Graph-I: Analysis of the different model metrics.

C. Confusion Matrix Analysis

As a result of the evaluations of the confusion matrices, the advantages of the fusion model became apparent. While the overall rates of errors are low, the least impressive feature of this type of model was found to be the reduced rate of false negatives. Again, this represents a significant issue in phishing detection since false negatives may place users on a direct path to being scammed. While it may seem difficult to assign a priority list to terms like "false negatives" when looking for a new phishing site, the practical effects of providing a large increase in the number of legitimate phishing sites is likely to provide a considerable advantage to users. Table-2 provides a tabular representation of the summary of confusion matrices for the test data set.

TABLE II: CONFUSION MATRIX SUMMARY

Model	TP	TN	FP	FN	Legit Precision(%)	Phishing Recall(%)
Fusion	20,163	14,991	151	65	99.01	99.68
Random Forest	20,188	14,993	149	40	99.01	99.8
XGBoost	20,163	14,935	207	65	98.63	99.68
CNN	80	20	11	10	64.01	88.57

VI. CONCLUSION AND FUTURE WORK

This project demonstrated a multi-modal phishing detection model that fused allusion to classically-trained machine learning, CNN-based visual deception detection, and GNN-based structural reasoning using an attention-based approach to combining input from these modalities to achieve state-of-the-art accuracy levels with reasonable inference speed based on the 230,000 sample training set. This multi-modal phishing detection system has combined many forms of digital evidence including technology with traditional methods such as classical machine learning, CNN VS GNN logical analysis and all were combined with an attention-based fusing method. This method has lead to the best model developed by a machine to reach state of the art performance with respect to accuracy yet has maintained a reasonable speed for inference time. The data used for the analysis included a combination of approximately 230,000 samples which adds some degree of confidence that the result is not simply a case of small data set anomaly.

A. Summary of Contributions

This effort led to the creation of a large-scale multi-modal phishing detection dataset that combines matching tabular URL data, web image capture and HTML DOM graph of the respective websites to allow for complete evaluation of phishing behaviour based on the amassed data. In addition, three complementary detection approaches (branches) were developed and implemented. The first was a classical ML application, which uses textual invoking and network information as input for deducing whether an incoming URL may be a phishing

site, as established by the dataset; the second is a CNN model that identifies whether a graphical representation of a website conveys an image of deception; and the last used a GNN model to reason about the structure of a site (HTML) captured in the dataset in conjunction with the other two branches to aid in identifying incoming requests to the web application as being malicious or not.

The creation of a true-to-scale, comprehensive category of phishing dataset that includes all elements of digital phishing from a given sample. Each sample contains: 1) a set of tabular URL data, 2) the corresponding image of the web page and 3) the corresponding HTML DOM graph. These elements will enable the machine to be able to identify phishing attempts from a combination of features and/or digital tokens such as a URL and image in conjunction with the logical structure of HTML.

B. Key Findings

The reason why phishing sites are difficult to detect is because they look similar to legitimate websites (e.g., logos, layout, colour schemes). The first finding in this study is that phishing sites have tabular features which will help to develop an initial baseline of features to detect that a site is a phishing site in the first instance. The second finding is that CNNs are able to identify visual deception, such as the difference between a fake and a real website. The third finding is that GNNs can identify structural anomalies on a given web page.

C. Future Research Directions

Our future research will focus on developing transformer-based models for evaluating adversarial robustness and temporal phishing analysis. In addition, future work will consider browser-level deployment of models, development of frameworks for continual learning and Federated Learning, as well as developing improved explainability methods for AI-based Phishing Detection.

REFERENCES

[1] Nayak, G. S., Muniyal, B., & Belavagi, M. C. (2025). Enhancing phishing detection: a machine learning approach with feature selection and deep learning models. IEEE Access.

[2] Salahdine, F., El Mrabet, Z., & Kaabouch, N. (2021, December). Phishing attacks detection a machine learning-based approach. In 2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON) (pp. 0250-0255). IEEE.

[3] Chinnasamy, P., Kumaresan, N., Selvaraj, R., Dhanasekaran, S., Ramprathap, K., & Boddu, S. (2022, November). An efficient phishing attack detection using machine learning algorithms. In 2022 International Conference on Advancements in Smart, Secure and Intelligent Computing (ASSIC) (pp. 1-6). IEEE.

[4] Almseidin, M., Zuraiq, A. A., Al-Kasassbeh, M., & Alnidami, N. (2019). Phishing detection based on machine learning and feature selection methods.

[5] Nataraj, K. R., Yashaswini, D. K., Hema, R., Pawar, N. S., & Yashaswi, S. (2022, December). Phishing attack detection using machine learning. In International Conference on Data Science, Machine Learning and Applications (pp. 355-370). Singapore: Springer Nature Singapore.

[6] Chinnasamy, P., Kumaresan, N., Selvaraj, R., Dhanasekaran, S., Ramprathap, K., & Boddu, S. (2022, November). An efficient phishing attack detection using machine learning algorithms. In 2022 International Conference on Advancements in Smart, Secure and Intelligent Computing (ASSIC) (pp. 1-6). IEEE.

[7] Gupta, R. (2020, February). Feature selection techniques and its importance in machine learning: a survey. In 2020 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS) (pp. 1-6). IEEE.

[8] Ye, H. J., Liu, S. Y., Cai, H. R., Zhou, Q. L., & Zhan, D. C. (2024). A closer look at deep learning methods on tabular datasets. arXiv preprint arXiv:2407.00956.

[9] Naseeb, S., Ramzan, S., Raza, A., Hashmi, M. S. A., Gu, Y., Syafrudin, M., & Fitriyani, N. L. (2025). Website Phishing Attack Detection Using Innovative Meta Learning Based Ensemble Approach. IEEE Access.

[10] Ahmed, A. A., & Abdullah, N. A. (2016, October). Real time detection of phishing websites. In 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON) (pp. 1-6). IEEE.

[11] Sahingoz, O. K., Buber, E., Demir, O., & Diri, B. (2019). Machine learning based phishing detection from URLs. Expert Systems with Applications, 117, 345-357.

[12] Bergholz, A., Paaß, G., D'Addona, L., & Dato, D. (2010). A real-life study in phishing detection. In Proceedings of the conference on email and anti-spam (CEAS) (Vol. 1, pp. 1-10).

[13] Linh, D. M., Hung, H. D., Chau, H. M., Vu, Q. S., & Tran, T. N. (2024). Real-time phishing detection using deep learning methods by extensions. International Journal of Electrical and Computer Engineering (IJECE), 14(3), 3021-3035.

[14] Asiri, S., Xiao, Y., Alzahrani, S., & Li, T. (2024). PhishingRTDS: A real-time detection system for phishing attacks using a Deep Learning model. Computers & Security, 141, 103843.

[15] Zaimi, R., Hafidi, M., & Lamia, M. (2023). A deep learning approach to detect phishing websites using CNN for privacy protection. Intelligent Decision Technologies, 17(3), 713-728.

[16] Alorvor, M. L. E., & Dadkhah, S. (2025). Real-Time Phishing Detection for Brand Protection Using Temporal Convolutional Network-Driven URL Sequence Modeling. Electronics, 14(18), 3746.
[17] Lindamulage, J. M., MandiraPabasari, L., Yapa, S. P. J., Perera, I. S. S., & Krishara, J. (2023, December). Vision gnn based phishing website detection. In 2023 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES) (pp. 1-7). IEEE.
[18] Shakir, S. S., Mohammad Khanli, L., & Emami, H. (2025). Convolutional graph network-based feature extraction to detect phishing attacks. Future Internet, 17(8), 331.
[19] Hofmans, W., Wei, W., Vanneste, S., & Mets, K. Towards Efficient GNN-Based Phishing Detection from HTML Source Code. In The 37th Benelux Conference on Artificial Intelligence and the 34th Belgian Dutch Conference on Machine Learning.
[20] Wang, C., Liu, Z., & Zeng, Y. (2024, August). Detecting Phishing URLs with GNN-based Network Inference. In 2024 International Conference on Networking and Network Applications (NaNA) (pp. 504-509). IEEE.